Creating the Document of Software Development Quality Assurance of School Social Network (SSN) based on IEEE 730-2002 Standard

Achmad Holil Noor Ali, Ika Nurkhasanah, Hanim Maria Astuti
Department of Information Systems, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember

Keywords:
Software development methodology
Software quality assurance
Best practice
IEEE 730-2002 standard

ABSTRACT

Ideally, the development of an application should follow a software development methodology in order to provide a clear direction of the development process that ultimately will yield a high quality software product. However, many cases happen when the development of software is failed to adopt any software methodology and in addition to that, absence to consider the standard used for assuring the quality of software. These lead to some problems: 1) the control to the software quality assurance was too difficult to conduct, 2) the deviation of requirements, 3) error founds during validation and verification processes and 4) the difficulty for future development. All this time, there is no such comprehensive instrument available to help developers ensure the quality of software development process. This study is intended to identify a document and instruments required by a developer team in assuring the quality of software during development phase. To identify the document and instruments, we first need to find out the software development methodology of an application and then adjust the software methodology with the IEEE 730-2002 standard. To do so, we use School Social Network (SSN) as our case study. This study contributes to complete literature on software quality assurance that currently is still absence to provide a comprehensive literature about document and instruments used for developers.

Corresponding Author:

Achmad Holil Noor Ali
Department of Information Systems
Faculty of Information Technology,
Jalan Raya ITS Kampus Sukolilo, Surabaya, Indonesia
Email: holil@its-sby.edu

1. INTRODUCTION

School Social Network (SSN) is a social network application developed by Al-Azhar Foundation on Education. This application serves as a means to facilitate the communication among teachers, students and the managerial of schools. There are two types of SSN version: mobile application and web based application.

By the time this study is being conducted, the development of SSN is in the testing phase. However, the developer of this application encounters several problems related to lack of control to ensure the quality of the application. Having an interview with them, this problem was occurred due to: 1) the software was developed without following any software development methodology. There was no clear direction as well as appropriate flows to develop the software so the team was difficult to control the software project. As a result, the scope of project was then bigger and thus, the project is overtime and over budget; 2) the software developer team did not utilize any software quality infrastructure (such as templates, checklists, procedures, work instructions, et cetera) during the software development process. Several drawbacks were then appeared: a) deviation to software requirements since there is no uniformity in procedure and work instruction for each development activity, b) during verification and validation to client, there were many errors found. This was the result of absent in using checklist to document and detect erroneous of software since the very beginning and c) difficulty to reengineer and maintain SSN since there is no document that guides the development and maintenance process.

Those problems are likely encontered by many software developers. This study contributes to overcome those aforementioned problems. This study aims to identify a document and instruments required by a developer team in assuring the quality of software during development phase. The idenfitication is based on IEEE 730-2002 [6] and software development methodology. The result of this study is expected to help the software development team of SSN control the future development of SSN. In addition, the result of this study can also be a guideline for other software development teams in developing their software.

2. LITERATURE REVIEW

2.1 Software Development Methodology

Software development methodology is a methodology used to plan, structure and control software development [9]. Choosing a right software development methodology is necessary to help software developers deliver projects on time, within the budget while meeting the requirements of customers. The software methodology itself covers techniques, tools, procedures, standards and documentations that help software developers develop software.

There are many types of software development methodology. However, there is no project that suits all of methodologies. Instead, choosing the right methodology that fits the nature of the project is of important. Several factors that influence the selection of software methodology are [2]: number of team members, the level of changes that will be occurred, the aim and the scope of a project, and the intensity of communication and relationship between developers and clients.

The methodology to develop software development is classified into two categories, traditional and Agile methodology. There is a main difference between these two categories, according to [3]. Traditional methodology focuses on completing requirements before moving into the next stage of software development, that is, design. The other methodology, agile methodology, promises a quick adaptation to collaboration, changes, process, tools, contract and planning.

Many models to develop software are categorized into either traditional, agile or perhaps the combination of both. Waterfall model, V-model, prototype model are the common methods of traditional methodology of software development. Agile methodology also comprises of many models. Some of the prominent models are Extreme programming (XP Model) and Scrum.

2.2 Software Quality Assurance

According to IEEE [2], software quality assurance is a set of planned activities aimed to produce software that complies with requirements. In addition to that, software quality assurance is a set of set of activities intended to evaluate software development process. In a project to develop software, software quality assurance is conducted in the entire project life cycle, started from initiation, planning, execution, closing and monitoring and controlling.

In his book, Galin [1] proposes architecture of software quality assurance that consists of 3 components: 1) pre-project SQA components. These consist of activities to ensure the quality of software during project initiation and planning process, 2) project life cycle SQA components that consist of activities to ensure the quality of software during software development process, and 3) quality infrastructure components that consist of tools and practices required to control and monitor software during initiation, planning, development and monitoring and controlling process, 4) quality management components that are used to ensure the software quality during monitoring and controlling process, 5) standards components. These components consist of standards used to ensure the quality of software in the entire project life cycle, and 6) organizational or human components. These activities are also conducted to ensure the quality of software in the entire project life cycle.

Among other components, the nature of quality infrastructures components is common to many projects. These components are infrastructure in a project and they serve as main tools used to prevent erroneous of software and to keep the software in certain level of quality. Some types of infrastructure components are procedure and work instruction, templates, checklists, preventive and correction actions, software configuration management, et cetera.

In this study, we identify a document and some infrastructure components that are commonly used in many projects. The infrastructures are checklist, procedures, and work instructions.

2.3 Software Quality Assurance Standards

IEEE 730-2002 is one of standards used in software quality assurance. Some tasks included in this standard [6] are the evaluation of system requirement and analysis, the evaluation of software and system design, media specification, the evaluation of unit implementation, integration and software testing. The

IEEE 730-2002 only specifies the above points. However, the standard is still absent on the output or deliverable of each point. In order to determine the output of each point, other standards are required such as IEEE Std. 829-1998 *System & Software Test Documentation* as the deliverable of software testing process [7] and ISO IEC 12207 / IEEE 12207-2008 as the deliverable of requirement gathering and analysis phase which is usually called Software Requirement Specification document [8].

3. METHODOLOGY

In order to identify instruments used to ensure the quality of software development, we use this following methodology.

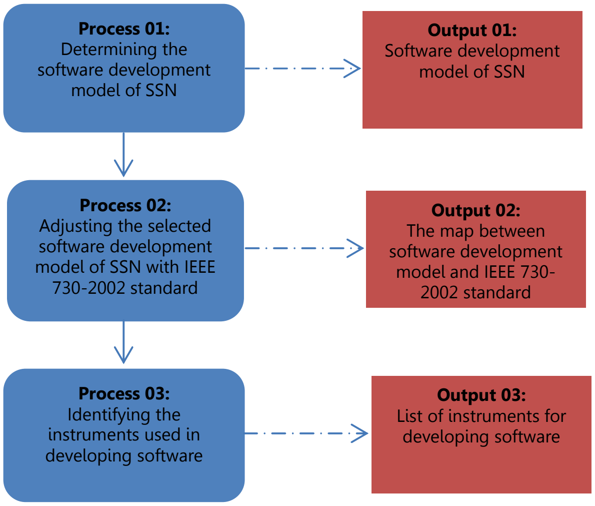
Process 01: Determining the software development model of SSN
Output 01: Software development model of SSN
Process 02: Adjusting the selected software development model of SSN with IEEE 730-2002 standard
Output 02: The map between software development model and IEEE 730-2002 standard
Process 03: Identifying the instruments used in developing software
Output 03: List of instruments for developing software

Figure 1. Methodology of the study

4. RESULTS AND ANALYSIS

In this chapter, we present the result of the each process in the methodology.

4.1 Determining the Software Development Model of SSN

Determining the software development model of SSN was started by an interview with development team, sponsor and the client of SSN. The interview result is used as a base to classify the most appropriate methodology of the SSN. We then compare the review result with the characteristics of each software development model written by [3][4].

Table 1. Selecting the software method for SSN

Selection Factors [3]	The Characteristics of SSN Development	Traditional	*Agile*
Number of developers	Less than 10 people		√
Changes during the project	Level of changes is high		√
Project objective	Supporting application emphasizing on system security	√	
	Requires rapid development time		√
Requirement management	Requires *baseline*	√	
	Flexibility to encounter any change anytime		√
Communication	Intensive communication and most likely face-to-face communication		√
Customer relationship	Requires a contract that clearly defines the relationship between developers and the client	√	
Organizational culture	Unspecific, only project manager and programmer which have a clear role definition		√

Table 1 shows the result of our comparison in selecting software method for SSN considering selection factors taken from [3]. According to the checklist, the most appropriate software method for SSN is agile methodology. However, there are many models included as Agile method. So, the next stage is to determine which Agile model that suits the nature of SSN.

Table 2. Selecting the most suited Agile model

Software Development Models	Key Points	Special Features
***Extreme Programming* (XP) [5]**	**The development of software is oriented to customers, small team, daily builds.**	**Refactoring. The process of system redesign is conducted to improve the performance and respond to changes.**
***Features Driven Development* (FDD)**	The process consists of 5 steps, object based components, **short iteration**.	**Simple method, design and system implementation based on features**, *object modeling*.
Scrum	Independent, **small project, self-organizing team**, *30-day release cycle*.	**Emphasizing on the paradigm of "definition and repetition" in the new product development.**

Table 2 shows how we select the Agile model for SSN. We compare the key points and special features of three models, i.e. Extreme Programming (XP), Features Driven Development (FDD) and Scrum. The bold and underlined key points and features indicate the suitability with the nature of SSN. Considering the comparison of the three models, the most suited Agile model for SSN is Extreme Programming.

4.2 Adjusting the Selected Software Development Model of SSN with IEEE 730-2002 Standard

The next stage after having Extreme Programming as the software development model is to adjust the model with the IEEE 730-2002 standard. The standard comprises of all tasks and activities applied for all models. Since SSN tends to use Extreme Programming, we need to adjust the IEEE 730-2002 with the processes and practices of assuring quality in Extreme Programming.

Each activity requires deliverables. However, there is no such document explained in the IEEE 730-2002. Therefore, we require other standards as our guideline to define the deliverables as well as documents. The related standards are IEEE Std. 829-1998 (System & Software Test Documentation) and ISO IEC 12207-2008/IEEE 12207-2008 (System & Software Test Requirement Specification). The tasks, activities, input and also the output are presented in Table 3.

4.3 Identifying the Instruments Used in Developing Software

The document of software development quality assurance which is the output of this study consists of two parts: 1) Main document and 2) supporting document. Main document is the guideline document to ensure the quality of software development of the SSN. The creation of this document refers to IEEE 730-2002 standard. However, we need a supporting document in order to conduct the activity of assuring the quality of software development of SSN such as procedures, checklist, work instruction, form and template. Table 3 shows an example of tasks, activities, input and output and also the related documents required in one Extreme Programming phase, that is, exploration phase.

Table 3. Tasks, activities and related documents required for developing SSN in Exploration Phase

The software development process of SSN (using XP)	**Quality Assurance Task of SSN (adjusting IEEE 730-2002 and XP)**	**Quality Assurance Activities (adjusting IEEE 730-2002 and XP)**	**Input and Output**	**Related Quality Assurance Standard**	**The Document of Software Development Quality Assurance of SSN**
Exploration phase	**Evaluation on exploration phase**				
	Evaluation on requirement elicitation				**Procedure:** [PR-01 R00] Evaluation Procedure of Requirement Elicitation of SSN **Policy:** [KE-01 R00] Requirement Elicitation
		Verification that participations are involved in determining system	**Input:**		
			Discussion result about job description of each element of SSN	-	**Form:** [FM-02 R00] Discussion on role/responsibilities of each element in developing SSN

The software development process of SSN (using XP)	**Quality Assurance Task of SSN (adjusting IEEE 730-2002 and XP)**	**Quality Assurance Activities (adjusting IEEE 730-2002 and XP)**	**Input and Output**	**Related Quality Assurance Standard**	**The Document of Software Development Quality Assurance of SSN**
		requirements	development		
			Responsibilities matrix	IEEE Std, 730-2002 *Software Quality Plan* • Annex: Responsibility Matrix	**Annex A Responsibility Matrix** [TE-01 R00] The Specification of Requirement of SSN (SRS of SSN).
			Output:		
			System Requirements Analysis Process Audit Checklist	Figure B-3. IEEE Std 730-2002	***Checklist*** : [CH-01 R00] *Checklist* Evaluation on Requirement Elicitation of SSN

Extreme programming consists of four phases: exploration, planning, iteration to release, productionizing, maintenance and death phase. Due to space limitation, we only present the identification of instrument for assuring software development in one phase only, i.e. exploration phase. However, the process of identifying tasks, activities, input, output and the document in other phases is similar to the exploration phase.

5. CONCLUSION

It is necessary for software developers to develop their software following the appropriate software development models that fit the nature of their project. Maintaining the quality of software development with regard to certain software development model is a must but a difficult task to do. IEEE 730-2002 provides a comprehensive set of tasks and activities to assure the quality of software development applied for all software development models. However, this standard is still absent in providing required documents and also instruments to assure the quality of software whereas these kinds of documents and instruments are necessary for software developers.

In this study, we begin with a problem where the software developers of our case study, SSN, have several challenges in developing the software. The challenges appear due to inability to define which software development model that fits the nature of SSN. As a result, the practice to assure the quality of software development encounters some problems. This study reveals that the most suited software development model for SSN is Agile methodology. More specific than that, the development tends to adopt an Extreme Programming model, one of Agile methodologies. Having known the software development model, an observation was conducted to adjust the IEEE 730-2002 standard with Agile model. Document and related instruments to ensure the quality of software development are then identified based on the mapping between Agile model and the standard. The document and related instruments can be used as a guideline for software developers of SSN to develop the next version of SSN or can be used for other developers of Extreme Programming model to develop their software.

The document and instruments in this study are only generated from our case study, which is limited to one software development model. For future research, it is interesting to complete the study with other software development models.

REFERENCES

[1] Daniel Galin, *“Software Quality Assurance From Theory to Implementation”*. London: Pearson Addison Wesley, 2004.

[2] Institute of Electrical and Electronics (IEEE), *"Software"*, IEEE Engineering Standards Collection, 1991.

[3] M. A. Awad, *"A Comparison between Agile and Traditional Software Development Methodologies”*, 2005.

[4] Hamid Faridani, *“A Guide to Selecting Software Development Methodology”*, 2011.

[5] Beck K, *“Embracing change with Extreme Programming”*, IEEE Computer Vol. 32, 1999.

[6] Institute of Electrical and Electronics (IEEE), *“IEEE 730-2002 Standard for Sofware Quality*

Assurance", Institute of Electrical and Electronics (IEEE), 2002.

[7] Institute of Electrical and Electronics (IEEE), *"IEEE Std. 829-1998 System & Software Test Documentation"*, Institute of Electrical and Electronics (IEEE), 1998.

[8] Institute of Electrical and Electronics (IEEE), *"ISO IEC 12207 / IEEE 12207-2008 Software Quality Management"*, Institute of Electrical and Electronics (IEEE), 2008.

[9] R.S. Pressman, *"Software Engineering – A Practitioner's Approach"*, European adaptation by D. Ince, 5th edn. London: McGraw-Hill International, 2000.

BIBLIOGRAPHY OF AUTHORS

	Achmad Holil Noor Ali is a Lecture in Information Systems Department, Institut Teknologi Sepuluh Nopember. In this department, he is a head of Planning and Development of Information Systems Laboratory. He received his B.Sc in Informatics from Institut Teknologi Sepuluh Nopember and a Master Degree in Computer Science in Universitas Indonesia. His research interests include information systems strategic planning, IT governance and IT project management. In addition to teaching, he has conducted some research projects funded by Institute of Research and Public Services, Institut Teknologi Sepuluh Nopember. He has also some papers published in national journal, national and international conferences.
	Ika Nurkhasanah was a bachelor student in Department of Information Systems, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember, Surabaya. This paper is one of her master pieces during her study in Information Systems department. During her study, she was also a very active student who had accomplished several national and international achievements. She is currently a junior consultant for a state-owned focusing in plantation that locates in Surabaya, East Java.
	Hanim Maria Astuti is currently an active member of the Planning and Development of Information Systems Laboratory in the Department of Information Systems, Institut Teknologi Sepuluh Nopember, Indonesia. She earned B.S. in Information Systems from Institut Teknologi Sepuluh Nopember, Indonesia; and M.Sc in Management of Technology from Delft University of Technology, the Netherlands. She has conducted some research projects granted by National Research Program, The Ministry of Higher Education, Indonesia, and has published conference papers. Her research interests include Mobile Service Innovation, Business Process Management and IT Quality Management.